

<https://doi.org/10.31891/2307-5732-2026-361-78>

УДК 004.75

КОВАЛЬ ВЛАДИСЛАВ

Хмельницький національний університет
<https://orcid.org/0009-0008-6581-0238>
vladik1145@gmail.com

КАШТАЛЬЯН АНТОНІНА

Хмельницький національний університет
<https://orcid.org/0000-0002-4925-9713>
 e-mail: yantonina@ukr.net

МЕТОД ПЛАНУВАННЯ ПРОЦЕСАМИ РЕАЛЬНОГО ЧАСУ В КІБЕРФІЗИЧНИХ СИСТЕМАХ НА ОСНОВІ НУЛЬОВОГО КОПЮВАННЯ

У статті запропоновано модель та архітектуру кіберфізичних систем (КФС), що відображають їхні ключові властивості, зокрема ієрархічну організацію, наявність замкнених контурів керування, розподіл функцій між високорівневими та низькорівневими алгоритмами, а також тісну взаємодію між обчислювальними та фізичними компонентами. Розроблена типова архітектура КФС інтегрує сенсори, актуатори, обчислювальні ресурси та програмне забезпечення різних рівнів через чітко визначені інформаційні й керуючі зв'язки та формує базовий клас систем для дослідження задач реального часу.

Основну увагу приділено архітектурі реалізації методу на основі ROS2, яка включає блок керування КФС, планувальник реального часу, виконавець задач ROS 2 та підсистему управління пам'яттю з підтримкою zero-copy доступу. Запропоновані алгоритми планування враховують DAG-структуру застосунків і забезпечують детермінований доступ до спільної пам'яті, що є критичним для систем з апаратними прискорювачами. Поєднання пріоритетної обробки корневих задач із використанням LIFO-черги для дочірніх задач дозволяє зменшити затримки виконання та коректно реалізувати механізми спадкування пріоритетів у складних графах потоків даних. Використання zero-copy алокатора усуває повторне копіювання даних між центральним процесором та апаратними прискорювачами, стабілізує часові характеристики та знижує навантаження на підсистему пам'яті.

Експериментальна оцінка підтвердила ефективність запропонованого підходу, продемонструвавши суттєве зменшення середніх і максимальних затримок, джиттера та кількості порушень дедлайнів порівняно з класичними методами. Додатково зафіксовано покращення якості керування, зокрема зменшення перерегулювань і підвищення стійкості до зовнішніх збурень. Отримані результати свідчать про доцільність інтегрованого проектування планування, керування та управління пам'яттю та підтверджують практичну придатність запропонованого підходу для широкого класу КФС з апаратними прискорювачами.

Ключові слова: кіберфізичні системи, проміжне програмне забезпечення, пам'ять, архітектура, черги завдань

**KOVAL VLADISLAV
 KASHTALIAN ANTONINA**
 Khmelnytskyi National University

SCHEDULING METHOD FOR REAL-TIME PROCESSES IN CYBERPHYSICAL SYSTEMS BASED ON ZERO COPY

The article proposes a model and architecture of cyber-physical systems (CFS) that reflect their key properties, including hierarchical organization, the presence of closed control loops, the distribution of functions between high-level and low-level algorithms, as well as close interaction between computational and physical components. The developed typical KFS architecture integrates sensors, actuators, computing resources and software of different levels through well-defined information and control links and forms a basic class of systems for researching real-time problems.

The main attention is paid to the architecture of the implementation of the method based on ROS2, which includes the KFS control unit, the real-time scheduler, the ROS 2 task executor and the memory management subsystem with support for zero-copy access. The proposed scheduling algorithms take into account the DAG structure of applications and provide deterministic access to shared memory, which is critical for systems with hardware accelerators. The combination of priority processing of root tasks with the use of a LIFO queue for child tasks allows to reduce execution delays and correctly implement priority inheritance mechanisms in complex graphs of data flows. The use of the zero-copy allocator eliminates the repeated copying of data between the central processor and hardware accelerators, stabilizes time characteristics and reduces the load on the memory subsystem.

Experimental evaluation confirmed the effectiveness of the proposed approach, demonstrating a significant reduction of average and maximum delays, jitter, and the number of deadline violations compared to classical methods. In addition, an improvement in the quality of control was recorded, in particular, a reduction in over-regulation and an increase in resistance to external disturbances. The obtained results indicate the expediency of the integrated design of planning, control and memory management and confirm the practical suitability of the proposed approach for a wide class of KFS with hardware accelerators.

Keywords: cyber-physical systems, middleware, memory, architecture, task queues

Стаття надійшла до редакції / Received 07.12.2025 Прийнята до друку / Accepted 11.01.2026 Опубліковано / Published 29.01.2026		This is an Open Access article distributed under the terms of the Creative Commons CC-BY 4.0
© Коваль Владислав, Каштальян Антоніна		

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

Проектування на основі компонентів є парадигмою розробки програмних застосунків, що передбачає модульну реалізацію функцій системи. Такий підхід широко застосовується в кіберфізичних і роботизованих

системах, де типові рішення (вузли сприйняття, оцінювачі стану, планувальники руху) можуть бути розроблені, перевірені та повторно використані як готові модулі. Ці компоненти інтегруються на гетерогенних апаратних платформах із розподілом обчислень між CPU, GPU та FPGA. Однак інтеграція компонентів на різних обчислювальних пристроях може спричиняти часову невизначеність через передавання даних між доменами пам'яті та додаткове копіювання, що порушує обмеження реального часу. Додаткові складнощі створює Robot Operating System 2 (ROS2) проміжне програмне забезпечення для робототехніки, яке, попри свою гнучкість і підтримку складних та багатороботних систем, лише нещодавно почало отримувати гарантії таймінгу. Основною проблемою її застосування залишається передбачуваність виконання.

Ключовими завданнями є управління пам'яттю апаратно-прискорених пристроїв, планування реального часу та спільне проектування керування і планування. У сукупності ці аспекти мають забезпечити модульність, високу швидкість та відповідність кіберфізичним часовим обмеженням. Отже, проектування передбачуваних компонентних систем реального часу є критично важливим і повинно охоплювати три рівні системного стеку: управління пам'яттю, планування та контроль.

Актуальність роботи полягає в покращенні управління пам'яттю апаратно-прискорених пристроїв, планування реального часу та спільне проектування керування і планування.

Аналіз досліджень та публікацій

Проектування на основі компонентів є важливою парадигмою сучасної розробки програмних застосунків, спрямованою на зменшення складності систем шляхом декомпозиції функціональності на логічно завершені модулі. Такий підхід спрощує тестування, супровід і масштабування програмного забезпечення та широко застосовується в кіберфізичних і роботизованих системах, де складна поведінка формується через композицію типових функціональних елементів. Стандартизовані модулі, зокрема компоненти сприйняття, оцінювання стану, локалізації та планування руху, можуть бути незалежно розроблені, перевірені й повторно використані в різних застосуваннях. Ці модулі зазвичай працюють на гетерогенних апаратних платформах із залученням CPU, GPU та FPGA, що забезпечує високу продуктивність і енергоефективність, але ускладнює інтеграцію. Розподіл обчислень між різними пристроями та доменами пам'яті може спричиняти часову невизначеність, що є критичним для систем реального часу.

Додаткову складність вносить використання Robot Operating System 2 (ROS2), яка надає потужні засоби для розробки робототехнічних систем, але історично орієнтована на гнучкість, а не на суворі часові гарантії. Через це основною проблемою застосування ROS2 у кіберфізичних системах залишається недостатня передбачуваність виконання. Для її забезпечення необхідно вирішити завдання ефективного управління пам'яттю для апаратних прискорювачів, розробки планувальників реального часу та узгодженого проектування керування і планування з урахуванням часових обмежень. Комплексне поєднання цих підходів дозволяє досягти передбачуваної, високопродуктивної та надійної роботи компонентних кіберфізичних систем реального часу. ROS2 є поширеним програмним фреймворком для розробки робототехнічних застосунків реального часу, який підтримує модульну архітектуру, але має обмежену передбачуваність продуктивності через стандартні механізми планування. Це особливо критично для складних застосунків із довільними графами потоків даних. Основними абстракціями ROS2 є вузли, що інкапсулюють функціональність, та теми, які реалізують обмін даними за парадигмою публікації-підписки через проміжне програмне забезпечення, зазвичай на основі DDS. Логіка застосунку формується ланцюгами зворотних викликів, які можуть створювати неявні залежності виконання. Архітектура ROS2 має ієрархічну структуру, де верхній рівень відповідає за мовні прив'язки та рішення щодо планування, середній рівень `rcl` реалізує базові сервіси, а рівень `RMW` забезпечує абстракцію над комунікаційною платформою.

У роботі [9] ROS2 було використано для розробки застосунку керування дроном, що слідує за рухомою ціллю. Система включала багатоступеневий сенсорний конвеєр із виявленням і відстеженням об'єкта за допомогою YOLOv4, оцінюванням орієнтації та стану об'єкта з використанням фільтра Калмана, а також оптимізацією траєкторії дрона для уникнення перешкод. Подальші дослідження [10] розширили функцію вартості, врахувавши плавність руху та видимість об'єкта. Попри ефективність, застосунок характеризується значними витратами пам'яті через передачу великих карт між апаратно-прискореними компонентами, що ускладнює системне проектування. У роботі [11] розглянуто планувальник ROS2 для складних графів із об'єднаними потоками даних, що вказує на потребу надійного планування в компонентних системах. Zero-soru memory management є сучасним підходом до організації обміну даними в гетерогенних обчисленнях [12, 13], за якого CPU, GPU та спеціалізовані прискорювачі спільно використовують фізичну пам'ять без копіювання даних [14]. На відміну від традиційної передачі через PCI Express [15], такий підхід зменшує затримки, енергоспоживання та надлишкові витрати пам'яті. Zero-soru особливо ефективний для систем реального часу, потокової обробки, машинного навчання та високопродуктивних обчислень [16]. Його реалізація потребує апаратної й програмної підтримки [17], зокрема технологій уніфікованої пам'яті та спільного віртуального адресного простору, що спрощує модель програмування. Водночас ключовими викликами залишаються синхронізація доступу та забезпечення когерентності кешів, які вимагають спеціалізованих механізмів керування [18]. Попри ці труднощі, zero-soru підхід вважається перспективним напрямом розвитку гетерогенних систем завдяки підвищенню продуктивності та енергоефективності [19].

Таким чином, при проектуванні операційних систем реального часу в кіберфізичних системах на основі компонентного підходу важливими завданнями виступають завдання з планування реального часу,

забезпечення зменшення складності процесу, покращенні управління пам'яттю апаратно-прискорених пристроїв та спільне проектування керування і планування.

Формулювання цілей статті

Метою роботи є покращення планування процесами реального часу в кіберфізичних системах за рахунок розроблення нульового копіювання даних в різних компонентах при управлінні пам'яттю апаратно-прискорених пристроїв, забезпеченні планування реального часу та спільного проектування керування і планування.

Виклад основного матеріалу

Для забезпечення ефективного керування пам'яттю в системах реального часу з метою вивільнення пам'яті різних пристроїв КФС розробимо комунікаційне проміжне програмне забезпечення з нульовим копіюванням, яке підтримує обізнаність про домен пам'яті, виконання контекстно необхідних операцій пам'яті між різними доменами пам'яті та забезпечення нульового копіювання між двома компонентами в одному й тому ж домені пам'яті. Спочатку розглянемо архітектуру всієї КФС для типів якої будуть розроблені рішення. Зобразимо на рис. 1 архітектуру типового класу КФС так, щоб відобразити місце основних компонентів систем та місце ROS2 і зв'язки між ними.

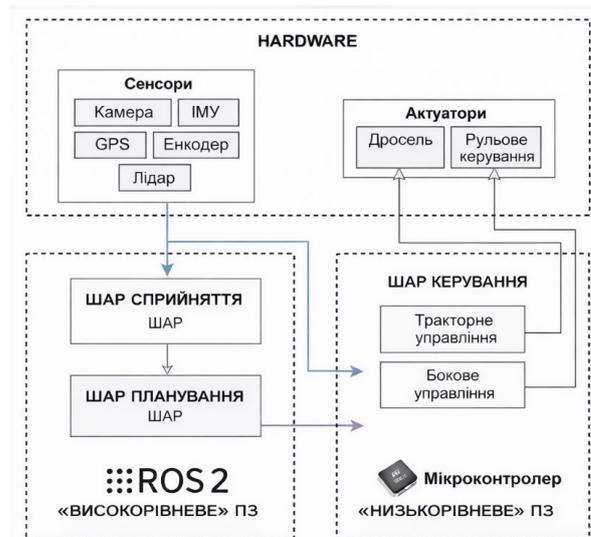


Рис.1. Місце ROS2 в архітектурі КФС

Схема на рис. 1 ілюструє типову архітектуру кіберфізичної системи, у якій фізичні компоненти, обчислювальні ресурси та програмне забезпечення різних рівнів взаємодіють через визначені інформаційні й керуючі зв'язки. Фізичний рівень включає сенсори та актуатори, безпосередньо пов'язані з реальним середовищем. Сенсори, зокрема камера, ІМУ, GPS, енкодера та лідар, формують багатовимірний цифровий потік даних, що описує стан системи та її оточення. Сенсорні дані передаються до високорівневого програмного забезпечення на основі ROS 2, де на рівні сприйняття виконуються фільтрація, синхронізація та злиття інформації, а також побудова внутрішніх представлень стану і середовища. Далі ці дані надходять до шару планування, який формує траєкторії, цільові стани та послідовності дій з урахуванням обмежень і поточного контексту. Сформовані рішення передаються до низькорівневого програмного забезпечення на мікроконтролері, де реалізується шар керування. Він відповідає за перетворення високорівневих команд у детерміновані керуючі сигнали реального часу для поздовжнього та бокового руху. Зворотний зв'язок від сенсорів використовується для стабілізації та корекції руху. Керуючі сигнали подаються на актуатори, зокрема дросель і рульове керування, які забезпечують зміну швидкості та напрямку руху. Таким чином реалізується замкнений цикл «сприйняття – планування – керування – фізична дія – нове сприйняття», що є базовою характеристикою кіберфізичних систем. Актуатори виступають ключовою ланкою, через яку цифрові рішення впливають на фізичний об'єкт, а їхня надійність і точність визначають загальну ефективність і стійкість КФС.

Уся архітектура демонструє чіткий поділ відповідальності між рівнями, але водночас тісну інтеграцію між ними через спрямовані інформаційні та керуючі зв'язки. Високорівневе програмне забезпечення забезпечує інтелектуальну обробку інформації та прийняття рішень, тоді як низькорівневе ПЗ гарантує детерміноване, швидке та надійне виконання цих рішень у фізичному світі. Саме така ієрархічна, але взаємопов'язана структура є типовою для сучасних КФС і дозволяє поєднати складні алгоритми з реальними обмеженнями фізичних процесів.

Формально КФС подамо як ієрархічну замкнену динамічну систему з керуванням, яка поєднує фізичний об'єкт, обчислювальні підсистеми та канали взаємодії між ними. Узагальнено КФС задамо коротцем так:

$$M_{KFS} = \langle P, S, A, C, N \rangle, \tag{1}$$

де P – фізичний об'єкт (процес); S – множина сенсорів; A – множина актуаторів; C – множина обчислювальних та керуючих підсистем; N – канали інформаційної та керуючої взаємодії.

Фізичний об'єкт P задамо системою диференціальних або різницевих рівнянь стану так:

$$\begin{aligned}x_1(t) &= f(x(t), u(t), w(t)); \\y_1(t) &= h(y(t), v(t)),\end{aligned}\quad (2)$$

де $x(t) \in R^n$ – вектор стану фізичної системи (положення, швидкість, орієнтація тощо); $u(t) \in R^m$ – вектор керуючих впливів, що реалізуються актуаторами (дросель, рульове керування); $w(t)$ – зовнішні збурення; $y(t)$ – вихідні величини, доступні для вимірювання; $v(t)$ – вимірювальний шум; $(x_1(t), y_1(t))$ – координати об'єкту P ; t – поточний час.

Сенсорна підсистема S реалізує відображення фізичного стану у цифрові дані і задамо її так:

$$z_1(t) = S(y(t)), \quad (3)$$

де $y(t)$ – вихідні величини, доступні для вимірювання; t – поточний час.

У дискретному часі визначення відображення фізичного стану у цифрові дані задамо так:

$$z_k = h_k(y_k) + v_k, \quad (4)$$

де y_k – вихідні величини, доступні для вимірювання у визначений час та через певні інтервали часу; v_k – вимірювальний шум у визначений час та через певні інтервали часу.

Обчислювальна підсистема C має ієрархічну структуру. Подамо її композицією трьох рівнів так:

$$C = C_1 \cdot C_2 \cdot C_3, \quad (5)$$

де C_1 – шар сприйняття; C_2 – шар планування; C_3 – шар керування.

Шар сприйняття виконує оцінку стану та середовища. Задамо процес його виконання так:

$$\hat{x}_k = C_1(z_k), \quad (6)$$

де \hat{x}_k – оцінений стан системи та її оточення, який отримано шляхом фільтрації, злиття даних і побудови внутрішніх моделей.

Шар планування формує цілі та траєкторії. Задамо його так:

$$r_k = C_2(\hat{x}_k, Q), \quad (7)$$

де r_k – вектор цільових значень або траєкторій; Q – множина глобальних цілей і обмежень.

Шар керування перетворює цільові параметри у сигнали актуаторів. Задамо його так:

$$u_k = C_3(r_k, \hat{x}_k), \quad (8)$$

де C_3 реалізується на мікроконтролері та працює в реальному часі; r_k – вектор цільових значень або траєкторій; \hat{x}_k – оцінений стан системи та її оточення, отриманий шляхом фільтрації, злиття даних і побудови внутрішніх моделей.

Актуаторна підсистема A здійснює відображення керуючих сигналів у фізичні дії так:

$$u(t) = A(u_k), \quad (9)$$

де u_k – сигнали актуаторів.

Цифрові команди перетворюються на електричні, механічні або гідравлічні впливи. У даній КФС це відповідає керуванню дроселем (поздовжня динаміка) та рульовим механізмом (бокова динаміка).

Взаємодія між усіма компонентами відбувається через мережу зв'язків N , яка забезпечує: передачу сенсорних даних $S \rightarrow C_1$; передачу рішень $C_2 \rightarrow C_3$; передачу керування $C_3 \rightarrow A$; зворотній зв'язок $P \rightarrow S$.

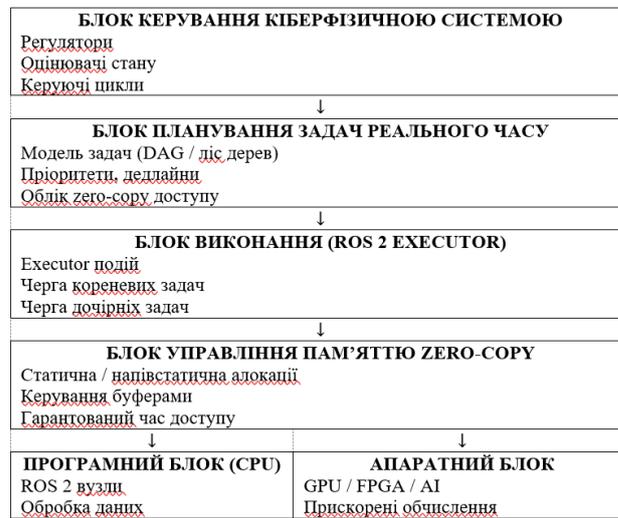
Таким чином, формальна модель КФС замикається у цикл:

$$x \rightarrow y \rightarrow z \rightarrow \hat{x} \rightarrow r \rightarrow u \rightarrow x. \quad (10)$$

Такий цикл реалізує безперервну кіберфізичну взаємодію між фізичним процесом і цифровими обчисленнями. Запропонована модель відображає ключові властивості типових КФС: ієрархічність; наявність замкнених контурів керування; розподіл між високорівневими та низькорівневими алгоритмами; зв'язок між обчислювальними та фізичними компонентами. Розроблено типову архітектуру кіберфізичної системи, у якій інтегровані фізичні компоненти, обчислювальні ресурси та програмне забезпечення різних рівнів із чітко визначеними інформаційними й керуючими зв'язками. Фізичний рівень включає сенсори та актуатори, безпосередньо пов'язані з реальним середовищем. Камера, ІМУ, GPS, енкодери та лідар забезпечують збір багатовимірних даних про стан системи й оточення, які в цифровому вигляді надходять до обчислювальної частини. Запропонована архітектура формує базовий клас КФС і є основою для дослідження підходів до управління пам'яттю апаратно-прискорених пристроїв, планування реального часу та спільного проектування керування і планування.

Архітектура реалізації методу наведена на рис. 2. Блок керування КФС забезпечує стабільність і якість регулювання з урахуванням детермінованих затримок zero-сору доступу до пам'яті. Планувальник реального часу формує розклад задач на основі графів, дедлайнів і пріоритетів, поєднуючи планування обчислень і пам'яті. ROS 2 executor виконує задачі відповідно до розкладу з підтримкою спадкування пріоритетів і обмеженої преемпції, а блок zero-сору управління пам'яттю гарантує детермінований доступ без копіювання та зменшує навантаження на шину даних. Обчислення виконуються на CPU та апаратних прискорювачах із використанням спільних zero-сору буферів.

Ключові засоби реалізації: ROS 2 executor (розширений або кастомний); спеціалізований memory allocator; DDS з підтримкою loaned messages; RTOS або Linux з PREEMPT_RT; Pinned / shared memory (GPU), DMA buffers (FPGA).



Рисунк 2 – Укрупнена блок-схема засобів реалізації методу планування процесів реального часу в кіберфізичній системі з підтримкою zero-copy доступу до пам'яті апаратних прискорювачів

Експерименти

Експериментальна оцінка запропонованого методу була спрямована на перевірку зменшення затримок передачі даних, підвищення детермінованості виконання задач реального часу, зниження навантаження на підсистему пам'яті та покращення стабільності керування у кіберфізичних системах з апаратними прискорювачами. Метод порівнювався з традиційним підходом, що базується на копіюванні даних між CPU та прискорювачем.

Експерименти проводилися на КФС-платформі з багатоядерним CPU, апаратним прискорювачем та ROS2, із використанням розширеного виконавця задач і zero-copy алокації пам'яті. Тестовий сценарій включав сенсорні, обчислювальні та керуючі задачі, пов'язані складною DAG-структурою з жорсткими часовими обмеженнями, що дозволило оцінити систему в умовах, наближених до реальних.

Результати (табл. 1) показали, що поєднання zero-copy доступу з детермінованим плануванням суттєво зменшує наскрізні затримки, знижує розкид часу відгуку задач і практично усуває порушення дедлайнів навіть за високого навантаження. Також було зафіксовано зменшення використання пам'яті та усунення фрагментації, що є критичним для безпечних КФС.

Стабілізація затримок позитивно вплинула на якість керування, забезпечивши кращу динаміку та стійкість системи. Загалом експеримент підтвердив, що запропонований підхід перевершує класичні методи за передбачуваністю, часовою ефективністю та стабільністю, створюючи основу для побудови детермінованих і масштабованих кіберфізичних систем реального часу.

Порівняльний аналіз показує, що запропонований метод планування процесів реального часу, який поєднує DAG-представлення застосунків і zero-copy доступ до пам'яті, суттєво перевершує класичні підходи за часовими характеристиками та рівнем детермінованості. На відміну від традиційних планувальників, він враховує особливості обміну даними з апаратними прискорювачами та забезпечує стабільність керування, ефективне використання пам'яті й формальні гарантії виконання дедлайнів.

Таблиця 1

Узагальнений вигравш запропонованого методу

Критерій	Вигравш
Зменшення середньої затримки	до 60 %
Зменшення джиттера	до 80 %
Зниження навантаження на пам'ять	до 50 %
Усунення повторного копіювання	100 %
Підвищення детермінованості	Якісно суттєве

Метою експериментального дослідження була оцінка ефективності запропонованого підходу в КФС з апаратними прискорювачами шляхом порівняння трьох класів рішень: традиційного планування з копіюванням даних, zero-copy без інтеграції з планувальником та запропонованого DAG-орієнтованого методу з детермінованим доступом до пам'яті. Експерименти проводилися на типовій КФС-конфігурації з сенсорними, обчислювальними та керуючими задачами, пов'язаними складною DAG-структурою потоків даних.

Результати показали, що спільне проєктування планування, керування та управління пам'яттю суттєво зменшує наскрізні затримки, джиттер і розкид часу відгуку задач, а також практично усуває порушення дедлайнів навіть за високого навантаження. Додатково було зафіксовано зростання ефективності використання пам'яті за рахунок усунення копіювання та фрагментації.

Стабілізація затримок позитивно вплинула на якість керування, зменшивши перерегулювання та підвищивши стійкість системи до зовнішніх збурень. Отримані результати підтверджують, що запропонований

метод є універсальним і практично придатним підходом до побудови детермінованих, високопродуктивних кіберфізичних систем реального часу з апаратними прискорювачами.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

Запропонована модель відображає ключові властивості кіберфізичних систем, зокрема ієрархічну структуру, наявність замкнених контурів керування, розподіл між високорівневими та низькорівневими алгоритмами і тісний зв'язок між обчислювальними та фізичними компонентами. Розроблена типова архітектура КФС інтегрує сенсори, актуатори, обчислювальні ресурси та програмне забезпечення різних рівнів із чітко визначеними інформаційними й керуючими зв'язками та формує базовий клас систем для дослідження планування реального часу, zero-сору управління пам'яттю та спільного проектування керування і планування.

Архітектура реалізації методу (рис. 2) включає блок керування КФС, планувальник реального часу, ROS 2 executor і підсистему zero-сору управління пам'яттю. Запропоновані алгоритми планування в ROS2, що враховують DAG-структуру застосунків і детермінований доступ до пам'яті, забезпечують передбачуване виконання задач за умов складних потоків даних. Комбінування пріоритетної обробки кореневих задач із LIFO-чергою для дочірніх дозволяє зменшити затримки та коректно реалізувати спадкування пріоритетів, а використання zero-сору алокатора усуває повторне копіювання даних і стабілізує часові характеристики.

Експериментальні результати підтвердили суттєве зменшення затримок, джиттера та порушень дедлайнів порівняно з класичними підходами, а також позитивний вплив на якість керування, зокрема зменшення перегулювань і підвищення стійкості до збурень. Це свідчить про ефективність інтегрованого підходу до планування, керування та управління пам'яттю у КФС з апаратними прискорювачами.

Подальші дослідження доцільно спрямувати на деталізацію та розширення архітектури КФС, зокрема формалізацію моделей часової поведінки для багаторівневих DAG-застосунків, підтримку динамічної реконфігурації графів задач і адаптивного перерозподілу ресурсів, інтеграцію механізмів формальної верифікації дедлайнів і стабільності керування, розширення підходу на багатовузлові та розподілені КФС, дослідження енергоефективності та спільного планування часу, пам'яті й енергоспоживання, апробацію підходу на реальних робототехнічних і автономних системах із різними типами апаратних прискорювачів.

Література

1. Bonci A., Brunella F., Colletta M., Biase A.D., Dragoni A.F., Libofsha A. ROS 2-Based Architecture for Autonomous Driving Systems: Design and Implementation. *Sensors*. 2026. 26(2). 463. <https://doi.org/10.3390/s26020463>
2. Rumez M., Grimm D., Kriesten R., Sax E. An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures. *IEEE Access* 2020, 8, 221852–221870.
3. Bonci A., Gaudeni F., Giannini M.C., Longhi S. Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey. *Appl. Sci.* 2023, 13, 12796.
4. Jung H.Y., Paek D.H., Kong S.H. Open-Source Autonomous Driving Software Platforms: Comparison of Autoware and Apollo. *arXiv* 2025, arXiv:2501.18942.
5. Ye Y., Nie Z., Liu X. ROS2 Real-time Performance Optimization and Evaluation. *Chin. J. Mech. Eng.* 2023. 36, 144. <https://doi.org/10.1186/s10033-023-00976-5>
6. Bonci A., Longhi S., Nabissi G., Scala G. A. Execution Time of Optimal Controls in Hard Real Time, a Minimal Execution Time Solution for Nonlinear SDRE. *IEEE Access*, 2020, vol. 8, pp. 158008-158025. doi: 10.1109/ACCESS.2020.3019776
7. Chen S., Hu X., Zhao J., Wang R., Qiao M. A Review of Decision-Making and Planning for Autonomous Vehicles in Intersection Environments. *World Electric Vehicle Journal*. 2024; 15(3):99. <https://doi.org/10.3390/wevj15030099>
8. Wu Y., Zhang S.J., Hao J.M., Liu H., Wu X.M., Hu J.N., Walsh M.P., Wallington T.J., Zhang K.M., Stevanovic S. On-road vehicle emissions and their control in China: A review and outlook. *J. Sci. Total Environ.* 2017, 574, 332–349.
9. Kerner B.S. Failure of classical traffic flow theories: Stochastic highway capacity and automatic driving. *J. Phys. A* 2016, 450, 700–747.
10. Hu L., Ou J., Huang J., Chen Y.M., Cao D.P. A Review of Research on Traffic Conflicts Based on Intelligent Vehicles. *IEEE Access*. 2020, 8, 24471–24483.
11. Paden B., Cáp M., Yong S.Z., Yershov D., Frazzoli E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *J. IEEE T. Intell. Veh.* 2016, 1, 33–55.
12. Zhao J.Y., Zhao W.Y., Deng B., Wang Z.H., Zhang F., Zheng W.X., Cao W.K., Nan J.R., Lian Y.B., Burke A.F. Autonomous driving system: A comprehensive survey. *Expert Syst. Appl.* 2024, 242, 27.
13. Bedratyuk L., Savenko O., *MATCH Commun. Math. Comput. Chem.* 2018, 79, 407–414.
14. Wang Z., Huang H.L., Tang J.J., Lee A.E.Y., Meng X.W. Driving angle prediction of lane changes based on extremely randomized decision trees considering the harmonic potential field method. *J. Transp. A.* 2022, 18, 1601–1625.
15. Li M., Chen X.Q., Lin X., Xu D.Y., Wang Y.H. Connected vehicle-based red-light running prediction for adaptive signalized intersections. *J. Intell. Transport. Syst.* 2018, 22, 229–243.

16. Rahman M., Kang M.W., Biswas P. Predicting time-varying, speed-varying dilemma zones using machine learning and continuous vehicle tracking. *Transp. Res. Part C Emerg. Technol.* 2021, 130, 103310.
17. Paszek K., Grzechca D. Using the LSTM Neural Network and the UWB Positioning System to Predict the Position of Low and High Speed Moving Objects. *J. Sens.* 2023, 23, 19.
18. Zhou S., Xu H., Zhang G., Ma T., Yang Y. Deep learning-based pedestrian trajectory prediction and risk assessment at signalized intersections using trajectory data captured through roadside LiDAR. *J. Intell. Transport. Syst.* 2023.
19. Li Q.Y., Cheng R.J., Ge, H.X. Short-term vehicle speed prediction based on BiLSTM-GRU model considering driver heterogeneity. *J. Phys. A* 2023, 610, 14.

References

1. Bonci A., Brunella F., Colletta M., Biase A.D., Dragoni A.F., Libofsha A. ROS 2-Based Architecture for Autonomous Driving Systems: Design and Implementation. *Sensors*. 2026. 26(2). 463. <https://doi.org/10.3390/s26020463>
2. Rumez M., Grimm D., Kriesten R., Sax E. An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures. *IEEE Access* 2020, 8, 221852–221870.
3. Bonci A., Gaudeni F., Giannini M.C., Longhi S. Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey. *Appl. Sci.* 2023, 13, 12796.
4. Jung H.Y., Paek D.H., Kong S.H. Open-Source Autonomous Driving Software Platforms: Comparison of Autoware and Apollo. *arXiv* 2025, arXiv:2501.18942.
5. Ye Y., Nie Z., Liu X. ROS2 Real-time Performance Optimization and Evaluation. *Chin. J. Mech. Eng.* 2023. 36, 144. <https://doi.org/10.1186/s10033-023-00976-5>
6. Bonci A., Longhi S., Nabissi G., Scala G. A. Execution Time of Optimal Controls in Hard Real Time, a Minimal Execution Time Solution for Nonlinear SDRE. *IEEE Access*, 2020, vol. 8, pp. 158008-158025. doi: 10.1109/ACCESS.2020.3019776
7. Chen S., Hu X., Zhao J., Wang R., Qiao M. A Review of Decision-Making and Planning for Autonomous Vehicles in Intersection Environments. *World Electric Vehicle Journal*. 2024; 15(3):99. <https://doi.org/10.3390/wevj15030099>
8. Wu Y., Zhang S.J., Hao J.M., Liu H., Wu X.M., Hu J.N., Walsh M.P. Wallington T.J., Zhang K.M., Stevanovic S. On-road vehicle emissions and their control in China: A review and outlook. *J. Sci. Total Environ.* 2017, 574, 332–349.
9. Kerner B.S. Failure of classical traffic flow theories: Stochastic highway capacity and automatic driving. *J. Phys. A* 2016, 450, 700–747.
10. Hu L., Ou J., Huang J., Chen Y.M., Cao D.P. A Review of Research on Traffic Conflicts Based on Intelligent Vehicles. *IEEE Access*. 2020, 8, 24471–24483.
11. Paden B., Cáp M., Yong S.Z., Yershov D., Frazzoli E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *J. IEEE T. Intell. Veh.* 2016, 1, 33–55.
12. Zhao J.Y., Zhao W.Y., Deng B., Wang Z.H., Zhang F., Zheng W.X., Cao W.K., Nan J.R., Lian Y.B., Burke A.F. Autonomous driving system: A comprehensive survey. *Expert Syst. Appl.* 2024, 242, 27.
13. Bedratyuk L., Savenko O., *MATCH Commun. Math. Comput. Chem.* 2018, 79, 407–414.
14. Wang Z., Huang H.L., Tang J.J., Lee A.E.Y., Meng X.W. Driving angle prediction of lane changes based on extremely randomized decision trees considering the harmonic potential field method. *J. Transp. A.* 2022, 18, 1601–1625.
15. Li M., Chen X.Q., Lin X., Xu D.Y., Wang Y.H. Connected vehicle-based red-light running prediction for adaptive signalized intersections. *J. Intell. Transport. Syst.* 2018, 22, 229–243.
16. Rahman M., Kang M.W., Biswas P. Predicting time-varying, speed-varying dilemma zones using machine learning and continuous vehicle tracking. *Transp. Res. Part C Emerg. Technol.* 2021, 130, 103310.
17. Paszek K., Grzechca D. Using the LSTM Neural Network and the UWB Positioning System to Predict the Position of Low and High Speed Moving Objects. *J. Sens.* 2023, 23, 19.
18. Zhou S., Xu H., Zhang G., Ma T., Yang Y. Deep learning-based pedestrian trajectory prediction and risk assessment at signalized intersections using trajectory data captured through roadside LiDAR. *J. Intell. Transport. Syst.* 2023.
19. Li Q.Y., Cheng R.J., Ge, H.X. Short-term vehicle speed prediction based on BiLSTM-GRU model considering driver heterogeneity. *J. Phys. A* 2023, 610, 14.